ON p-ADIC CLASSIFICATION

PATRICK ERIK BRADLEY

Abstract. A p-adic modification of the split-LBG classification method is presented in which first clusterings and then cluster centers are computed which locally minimise an energy function. The outcome for a fixed dataset is independent of the prime number p with finitely many exceptions. The methods are applied to the construction of p-adic classifiers in the context of learning.

1. Introduction

The field $\mathbb{Q}_p$ of p-adic numbers is of interest in hierarchical classification because of its inherent hierarchical structure [10]. A great amount of work deals with finding p-adic data representation (e.g. [8, 9]).

In [4], the use of more general p-adic numbers for encoding hierarchical data was advocated in order to be able to include the case of non-binary dendrograms into the scheme without having to resort to a larger prime number p. This was applied in [5] to the special case of data consisting in words over a given alphabet and where proximity of words is defined by the length of the common initial part. There, an agglomerative hierarchic p-adic clustering algorithm was described. However, the question of finding optimal clusterings of p-adic data was not raised.

Already in [1], the performance of classical and p-adic classification algorithms was compared in the segmentation of moving images. It was observed that the p-adic ones were often more efficient. Learning algorithms using p-adic neural networks are described in [2, 6].

Inspired by [1], our main concern in this article will be a p-adic adaptation of the so-called split-LBG method which finds energy-optimal clusterings of data. The name "LBG" refers to the initials of the authors of [7], where it is described first. Their method is to find cluster centers, and then to group the data around the centers. In the next step, the cluster centers are split, and more clusters are obtained. This process is repeated until the desired class number is attained. For p-adic data, this approach does not make sense: first of all, cluster centers are in general not unique; and secondly, because the dendrogram is already determined by data, an arbitrary choice of cluster centers is not possible—this can lead to incomplete clusterings. Hence, we first find clusterings by refining in the direction of highest energy reduction, until the class number exceeds a prescribed bound. Thereafter, candidates for cluster centers are computed: they minimise the cluster energy. The result is a sub-optimal method for p-adic classification which splits a given cluster into its maximal proper subclusters. A variant discards first all quasi-singletons, i.e. clusters of energy below a threshold value. The *a posteriori* choice of centers turns out useful for constructing classifiers.

Date: June 22, 2009.

A first application of some of the methods described here to event history data of building stocks is described in [3]. There, the classification algorithm is performed on different p-adic encodings of the data in order to compare the dynamics of some sampled municipal building stocks.

After introducing notations in Section 2, we briefly describe the classical split-LBG method in Section 3. Section 4 reformulates the minimisation task of split-LBG in the p-adic setting, and describes the corresponding algorithms. The issue on the choice of the prime p is dealt with in Section 5. Section 6 constructs classifiers and presents an adaptive learning method in which accumulated clusters of large energy are split.

2. Generalities

2.1. **p-adic numbers.** Let p be a prime number, and K a field which is a finite extension field of the field $\mathbb{Q}_p$ of rational p-adic numbers. We call the elements of K simply p-adic numbers. K is a normed field whose norm $|\ |_K$ extends the p-adic norm $|\ |_p$ on $\mathbb{Q}_p$. Let $\mathcal{O}_K := \{x \in K \mid |x|_K \le 1\}$ denote the local ring of integers of K. Its maximal ideal $\mathfrak{m}_K = \{x \in K \mid |x|_K < 1\}$ is generated by a *uniformiser* π. It has the property $v(\pi) = \frac{1}{e}$, where $e \in \mathbb{N}$ is the ramification degree of $K/\mathbb{Q}_p$.

All elements $x \in K$ have a π-adic expansion

$$x = \sum_{i \ge -m} \alpha_i \pi^i \tag{1}$$

with coefficients α_i in some set $\mathcal{R} \subseteq K$ of representatives for the residue field $O_K/\mathfrak{m}_K \cong \mathbb{F}_{p^f}$. In the case $q = p$, the choice $\mathcal{R} = \{0, 1, \dots, p-1\}$ is quite often made.

By X will will always mean a finite set of data taken from K.

2.2. **p-adic clusters.** A *disk* in some finite set $X \subseteq K$ is a subset of the form

$$\{x \in X \mid |x - a|_K < \varepsilon\}$$

for some $a \in X$ and $\varepsilon > 0$. In particular, any singleton $\{x\} \subseteq X$ is a disk in X.

The *cluster property* of a subset C of p-adic data $X \subseteq K$ is given by saying that for any $a \in C$ it holds true that

$$|x - a|_K < \mu(C) \Rightarrow x \in C, \tag{2}$$

where

$$\mu(C) := \max\{|x - y|_K \mid x, y \in C\}$$

is the cluster diameter. As a consequence, a cluster is a union of disks in X. We will call a disk in X also a *verticial cluster*, because in the in the dendrogram for X, the vertices correspond to those clusters which are (non-singleton)[1] disks. More to the dendrogram associated to p-adic data will be said in Section 4.1. In Figure 1 the ultrametric property of dendrograms is visualised as follows: data b, c connected by a path consisting of vertical and horizontal line segments are considered as near, if the sum of the vertical parts is short. A third datum a further away from b and c

[1] In many definitions of dendrograms, the data correspond to terminal vertices, but in our definition in Section 4.1, data are not considered as vertices of the dendrogram. Nevertheless, we do not exlude singleton clusters from the definition of "vertcial". We apologise for this inconsistency.

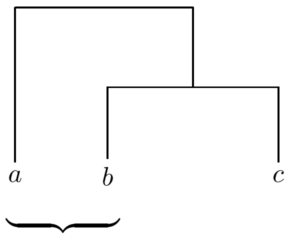

Figure 1. Dendrogram in which b, c are closer to each other than to a. It contains a subset which is not a cluster.

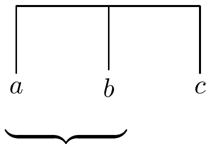

Figure 2. Dendrogram with equidistant data contains non-verticial clusters.

is, by ultrametricity, at equal distance to b and c. This fact is visualised by having paths $a \rightsquigarrow b$ and $a \rightsquigarrow c$ with vertical components summing up to equal length.

Example 2.1. *Let $X = \{a, b, c\}$, and consider the subset $C = \{a, b\}$. In Figures 1 and 2, we assume two different dendrograms for our data X. In Figure 1, the disks are the singletons, the set $\{b, c\}$, and the whole dataset X. Hence, C is not a cluster in the case of Figure 1, because it does not satisfy the cluster property (2): b and c are at distance less than the diameter which equals the distance between a and b, whereas C contains b but not c. However, in Figure 2, all data are at equal distance, so the only disks are the singletons and X. Hence, C is a cluster in Figure 2, but not a disk, i.e. not verticial.*

A *clustering* of X is a collection $\mathscr{C}$ of disjoint clusters of X whose union is the whole dataset X. It is called *verticial*, if it consists entirely of verticial clusters.

Notice that the definition of cluster depends on the dataset X. In particular, a non-verticial cluster can be made into a disk by deleting some data from X. E.g. in Figure 2 the removal of c from the dataset turns $C = \{a, b\}$ into a verticial cluster. In general, if $\mathscr{C}$ is a clustering of X, and $Y \subseteq X$, then $\mathscr{C}_Y := \{C \cap Y \mid C \in \mathscr{C}\}$ is the *restriction of $\mathscr{C}$ to Y*. This motivates us to consider only the case of verticial clusterings.

Assumption. *All clusterings we consider are verticial on some specified (non-empty) subsets of X.*

3. The split-LBG algorithm

Here, we review briefly the classical split-LBG algorithm. Details can be found in [7].

Let $X = \{a_1, \dots, a_n\}$ and $C = \{c_1, \dots, c_k\}$ be sets of vectors in $\mathbb{R}^m$, where X is considered as the *data* and C are the prespecified *cluster centers*. The task is classically to find a partition $\mathscr{O} = \{\Omega_c \mid c \in C\}$ of X into k clusters Ω_c minimising the energy

$$E(\mathscr{O}, C) = \sum_{c \in C} \sum_{a \in \Omega_c} d(a, c),$$

where $d(x, y)$ is Euclidean distance in $\mathbb{R}^m$. In fact, the split-LBG method works with varying C by alternatively constructing partitions and then replacing each $c \in C$ by two new centers $c + \varepsilon, c - \varepsilon$, where ε is a perturbation vector in $\mathbb{R}^m$ of small norm. From these, a new partition is constructed, etc.

4. Split-LBG in the p-adic case

In [1] it was observed that the split-LBG method has no direct translation using the p-adic metric. Here, we describe a p-adic modification of the task from the previous section.

Let $X = \{x_1, \dots, x_n\} \subseteq K$ be some data consisting of n p-adic numbers, and fix a number k. The task is to find a clustering $\mathscr{C} = \{C_1, \dots, C_\ell\}$ of X with $\ell \leq k$, and for each cluster $C \in \mathscr{C}$ a *center* $a_C \in C$, minimising the expression

$$E_p(X, \mathscr{C}, \mathbf{a}) := \sum_{C \in \mathscr{C}} \sum_{x \in C} |x - a_C|_K\,,$$

where $\mathbf{a} = (a_C)_{C \in \mathscr{C}}$ is the sequence of cluster centers.

Note that, by the ultrametric property of $|\;|_K$, cluster centers can (and will) always be chosen within X. This has already been taken care of in the definition of the task. Note further that, unlike in the Archimedean setting, cluster centers are in general not uniquely defined by their corresponding clusters.

The most significant difference to the Archimedean case is given by the fact that in the p-adic situation, it does not make sense to choose a cluster center *a priori*, as illustrated in Example 4.1. Therefore, the order is reversed: first find a good partition, and then find corresponding cluster centers.

Example 4.1. *Let $\{a, b, c\}$ be some data with corresponding dendrogram as in Figure 1. Then choosing a, b as centers leads to the clustering $\mathscr{C} = \{\{a\}, \{b, c\}\}$, whereas the choice b, c leads either to $\mathscr{C}' = \{\{b, c\}\}$, $\mathscr{C}'' = \{\{a, b, c\}\}$, or to $\mathscr{C}''' = \{\{b\}, \{c\}\}$. But $\mathscr{C}'$ and $\mathscr{C}'''$ are not clusterings of $\{a, b, c\}$, while $\mathscr{C}''$ is. And both $\mathscr{C}'$ and $\mathscr{C}''$ each consist of one cluster containing the two prescribed centers instead of two distinct clusters as should be the case classically.*

Last but not least, we will not give a global solution to the task in the p-adic case, but find certain types of local minima of E_p in a sense which will become clear in the following subsection.

4.1. **Some definitions.** An important tool in the classification of p-adic data $X \subseteq K$ is its dendrogram $D(X)$. In contrast to the Archimedean situation, it is uniquely determined by the data (cf. [4, 5]). We view $D(X)$ as a *rooted metric tree*. This means that it has a root v_0, and all edges are oriented away from v_0 and are assigned a length which is either positive real or infinite. The root v_0 corresponds to the top cluster consisting of the whole data X. The vertices correspond to clusters containing at least two points from X. An edge e of $D(X)$ connecting two vertices is always bounded. The individual points of X correspond uniquely to the *ends* of the tree $D(X)$. We do not view the data X as part of the tree $D(X)$, but as its boundary. Hence, any $x \in X$ sits at the one extreme of an unbounded edge. Our viewpoint is probably in contrast to most others on hierarchical classification, where data correspond to terminal vertices of dendrograms. However, we argue in our favour that the dendrogram should reflect hierarchic approximations of data by clusters (vertices in $D(X)$) or, more generally, by initial terms in some p-adic expansion for data (points in $D(X)$). We refer to [4, 5] for a more detailed description of p-adic dendrograms.

Given some vertex v of $D(X)$, let $\mathrm{ch}(v)$ denote the set of edges emanating from v (i.e. not towards v_0), and let $\#\mathrm{ch}(v)$ be its cardinality. By abuse of notation, we will identify $\mathrm{ch}(v)$ with the set of vertices and ends attached to the edges in $\mathrm{ch}(v)$.

Now, an upper bound for the contribution to E_p of a cluster C_v, represented by some vertex or end v is

$$\mu(v) := \mu(C_v) = \max\left\{\left|x-y\right|_K \mid x, y \in C_v\right\}.$$

As a side remark, note that this is nothing but the Haar measure of K evaluated in the p-adic disk $D_v \subseteq K$ corresponding to v. In any case, if v is an end then $\mu(v) = 0$, otherwise $\mu(v) > 0$.

Given a set V of vertices or ends of $D(X)$, we set

$$E(V) := \sum_{v \in V} (\#C_v - 1) \cdot \mu(v), \tag{3}$$

and also write $E(v_1, \dots, v_b)$ in the case that $V = \{v_1, \dots, v_b\}$. Applying this to $\mathrm{ch}(v)$ for a vertex v, we obtain:

$$E(\mathrm{ch}(v)) \leq E(v). \tag{4}$$

The following remark shows that minimising $E(V)$ does make sense for our task:

Remark 4.2. *Given a clustering $\mathscr{C} = \{C_v \mid v \in V\}$, where V is the corresponding set of vertices, for any choice of $\alpha_v \in C_v$ it holds true that*

$$E_p(X, \mathscr{C}, \mathbf{a}) \leq E(V) =: E(\mathscr{C}),$$

where $\mathbf{a} = (\alpha_v)_{v \in V}$.

Let $\mathfrak{X}_k(Y)$ be the set of all clusterings $\mathscr{C}$ of X with cardinality $\ell \leq k$ whose restriction to Y is verticial. On the set

$$\mathfrak{X} = \bigcup_{k \in \mathbb{N}} \bigcup_{Y \subseteq X} \mathfrak{X}_k(Y), \tag{5}$$

of all clusterings, we define a partial ordering $\leq$ (called *refinement*) as follows:

$$\mathscr{C}' \leq \mathscr{C},$$

if all $C \in \mathscr{C}$ are of the form $C = \bigcup\limits_{i \in I} C_i'$ with $C_i' \in \mathscr{C}'$ $(i \in I)$.

Let C_v be the smallest verticial cluster containing a given cluster C. Then we can define the functional

$$E \colon \mathfrak{X} \to \mathbb{R},\ \mathscr{C} \mapsto \sum_{C \in \mathscr{C}} (\#C - 1) \cdot \mu(C_v),$$

and observe that this obviously generalises $E(V)$ from (3):

Lemma 4.3. *If $\mathscr{C} \in \mathfrak{X}$ is verticial, then*

$$E(V) = E(\mathscr{C}),$$

where V is the vertex set associated to $\mathscr{C}$.

Lemma 4.4. *E is strictly monotonic:*

$$\mathscr{C}' \le \mathscr{C} \Rightarrow E(\mathscr{C}') \le E(\mathscr{C}),$$

and if $\mathscr{C}' \le \mathscr{C}$ are not equal, then $E(\mathscr{C}') < E(\mathscr{C})$.

Proof. Assume $C = \bigcup\limits_{i\in I} C_i' \in \mathscr{C}$ with $C_i' \in \mathscr{C}'$. Then

$$\sum_{i\in I} \#(C_i' - 1) \cdot \mu(C_{i,v}') \le \sum_{i\in I} \#(C_i' - 1) \cdot \mu(C_v) \le (\#C - 1) \cdot \mu(C_v),$$

where the first inequality holds true, because all C_i' are contained in C. The second inequality is strict, if I contains more than one element. That is the case for some C, if $\mathscr{C} \neq \mathscr{C}'$. □

We denote by $E_{k,Y}$the restriction of E to $\mathfrak{X}_k(Y)$. The following is immediate:

Lemma 4.5. *Let $\mathscr{C}$ and $\mathscr{C}'$ minimise $E_{k,Y}$ and $E_{k',Y}$, respectively. Then*

$$k \le k' \Rightarrow E(\mathscr{C}') \le E(\mathscr{C}).$$

4.2. **The verticial clustering algorithm.** The general strategy which we follow is to refine a given clustering of X in the "direction" which yields the lowest value of E_p after splitting a vertex. The term "direction" refers to the refinement ordering on $\mathfrak{X}$, and we follow the possible "gradients" from a given point $\mathscr{C} \in \mathfrak{X}$. Concretely, this means splitting a vertex with highest energy contribution. In Section 5, we will see that the terms in quotation marks here can be taken ad literam.

In this subsection, we deal with verticial clusterings only. We can now formulate:

Algorithm 4.6 (Verticial clustering)**.** *Input.* p-adic data $X \subseteq K$ with $\#X \ge 2$, and upper bound $k \ge 1$ for number of clusters.

Step 0. Compute $b = \#\mathrm{ch}(v_0)$ and $E(v_0) = \mu(v_0)$.

Step 1. If $b > k$, then terminate. Otherwise, compute $E(\mathrm{ch}(v_0))$ which is not greater than $E(v_0)$ by (4). Further identify the set of vertices $V_1 := \mathrm{ch}(v_0) \cap \mathrm{Vert}(D(X))$.

Step N. Assume that from the previous step, we are given some family $\mathscr{V}_{N-1} = \left\{V_{N-1}^{(i)}\right\}$ of sets consisting of $b_{N-1}^{(i)} \le k$ vertices, respectively. If for all i and all $v \in V_{N-1}^{(i)}$ it holds true that $b_v^{(i)} := b_{N-1}^{(i)} + \#\mathrm{ch}(v) > k$, then terminate.

Otherwise, find all i and all $v \in V_{N-1}^{(i)}$ such that $E(W_v^{(i)})$ is smallest possible, where $W_v^{(i)} := \mathrm{ch}(v) \cup V_{N-1}^{(i)} \setminus \{v\}$ satisfies $\#W_v^{(i)} \le k$. Again, by (4), it holds true that

$$E(W_v^{(i)}) \le E(V_{N-1}^{(i)}).$$

Extract this new family $\mathscr{V}_N$ of vertex sets together with the lower energy value $E_N = E(W)$ for $W \in \mathscr{V}_N$.

Output. A family of clusterings $\{\mathscr{C}_i \mid i \in I\}$ (corresponding to the vertex sets in the last step) for which $E = E(\mathscr{C})$ is locally minimal, together with the value of E.

4.3. **p-adic cluster centers.** The next objective is to find cluster centers with respect to the energy functional. Assume that we are given a fixed cluster $C = \{a_1, \dots, a_n\} \subseteq K$. We wish to find some $\alpha \in C$ which minimises

$$\epsilon(\alpha) := E_p(C, \mathscr{C}, \alpha) = \sum_{a \in C} |a - \alpha|_K\,,$$

where $\mathscr{C} = \{C\}$.

A *branch* B of a rooted tree (T, v) is a maximal subtree of $T \setminus \{v\}$. It has a root v_B among the vertices of $\mathrm{ch}(v)$. Let $\mathcal{B}(T)$ denote the set of branches of (T, v). In the case of our dendrogram $D(C)$, we will write $\mathcal{B}(C)$, instead of $\mathcal{B}(D(C))$. The branches induce a natural partition of C:

$$C = \bigcup_{B \in \mathcal{B}(C)} C_B$$

into a disjoint union of $C_B = \mathrm{Ends}(B)$.

Lemma 4.7. *Let $\alpha \in C$, and $B_\alpha \in \mathcal{B}(C)$ the branch containing α as an end, and $C_\alpha = C_{B_\alpha}$. Then*

$$\epsilon(\alpha) = \#(C \setminus C_\alpha) \cdot \mu(v_0) + E_p(C_\alpha, \mathscr{C}_\alpha, \alpha), \tag{6}$$

where $\mathscr{C}_\alpha = \{C_\alpha\}$.

Proof. Together with the identity:

$$\sum_{a \in C_\alpha} |a - \alpha|_K = E_p(C_\alpha, \mathscr{C}_\alpha, \alpha),$$

this follows easily by looking at the tree $D(C)$. □

Lemma 4.8. *Assume the notations as in Lemma 4.7. It holds true that*

$$\frac{\epsilon(\alpha)}{\mu(v_0)} = N_\alpha + O(p^{\nu_\alpha}) \tag{7}$$

with $N_\alpha \in \mathbb{N}$ and $\nu_\alpha < 0$.

Equation (7) means that $\frac{\epsilon(\alpha)}{\mu(v_0)}$ is a natural number plus some small term given as a multiple of p^{ν_α}.

Proof. Set $N_\alpha = \#(C \setminus C_\alpha)$, and notice that

$$E_p(C_\alpha, \mathscr{C}_\alpha, \alpha) \leq \#C_\alpha \cdot \mu(v_\alpha), \tag{8}$$

where v_α is the root of B_α. The claim now follows from the obvious inequality $\mu(v_\alpha) < \mu(v_0)$. □

Now, we can formulate our algorithm:

Algorithm 4.9 (Cluster centers)**.** *Step* 1. Find all branches $B^{(1)} \in \mathcal{B}(C)$ with largest value of $\#C_{B^{(1)}}$. Extract those clusters $C_{B^{(1)}}$ for which $\mu(v_{B^{(1)}})$ is minimal, and the number

$$c_1 = \max\left\{\#C_{B^{(1)}} \mid B^{(1)} \in \mathcal{B}(C)\right\}.$$

Step N. Assume that in the previous step, a list of clusters $C_{B^{(N-1)}}$, and a number c_{N-1} is produced. Find all branches $B^{(N)}$ of the rooted trees $D(C_{B^{(N-1)}})$ with largest possible value c_N of $\#C_{B^{(N)}}$. Extract those clusters $C_{B^{(N)}}$ minimising $\mu(v_{B^{(N)}})$, together with c_N.

At some point, there will be a *Step* N' in which the trees $D(C_{B(N)})$ have only one vertex each. The procedure terminates thus:

Output. A list $(C_i)_{i\in I}$ of those clusters from Step N' with minimal value of $\mu(v_i)$, where v_i is the vertex of $D(C_i)$.

Theorem 4.10. *Let $C' = C_{N'} \subseteq C$ be a cluster produced by performing Algorithm 4.9. Then any $\alpha \in C'$ is a center of C with respect to E_p.*

Proof. Let $C = C_0 \supsetneq C_1 \supsetneq \dots C_{N'} = C'$ be a strictly decreasing chain of clusters produced by the N' steps of Algorithm 4.9. Let the corresponding cardinalities be $c_0, \dots, c_{N'}$. By applying Lemma 4.7, it holds true that

$$\epsilon(\alpha) = c_{N'} \cdot \mu(v_{N'}) + \sum_{i=1}^{N'} (c_{i-1} - c_i) \cdot \mu(v_{i-1}), \tag{9}$$

where v_j is the root of the corresponding branch from *Step j*. The minimality of $\epsilon(\alpha)$ is guaranteed by (7), applied to each step. Notice, that we have used the obvious fact that for C', the inequality (8) is an equality. □

4.4. **Quasi-verticial clustering.** The two previous subsections already lead to a p-adic algorithm for verticial clusterings and their centers. In this case, subdividing a cluster C_v means to make as many subclusters as there are elements in $\mathrm{ch}(v)$. In the case that e.g. there are many singletons, this can be a disadvantage. Hence removing singletons provides more flexibility in that the bigger subclusters can either be merged or kept distinct. Even greater flexibility can be achieved if almost indistinguishable clusters are treated as singletons.

Definition 4.11. *Fix some real $\varepsilon > 0$. A verticial cluster $C_v \subseteq X$ with corresponding vertex v is called a* quasi-singleton *for ε, if $E(v) < \varepsilon$.*

When we speak of a quasi-singleton, we mean a quasi-singleton for some ε known from the context.

Example 4.12. *The dendrogram in Figure 3 contains a quasi-singleton $\{a, b\}$, if we set $\mu(v) = p^{-\ell}$ for vertex v at level ℓ (indicated by the number at the left), and $p^{-1} < \varepsilon \leq p^{-1}$. For this choice of ε, the cluster $\{c, d\}$ is not a quasi-singleton. But this is the case for larger ε.*

Clearly, any singleton is a quasi-singleton for any ε. Since we are working with a fixed p-adic field K, it is possible to choose ε so small that the quasi-singletons are precisely the singletons of our given dataset X.

The algorithm we propose in the following removes quasi-singletons in order to continue with verticial clusterings. For this, we fix some notation: When referring to a subset Y of our dataset X, we will indicate this by the subscript Y. E.g. $\mathrm{ch}_Y(v)$ means the set of edges in $D(Y)$ going out from v. Simliarly, with $\mu_Y(V)$, $E_Y(V)$ etc.

Algorithm 4.13 (Quasi-verticial clustering)**.** *Input.* Data $X_0 := X \subseteq K$, and numbers $k_0 := k \geq 1$, $\varepsilon > 0$.

Step 1. Remove from $D(X)$ all $v \in \mathrm{ch}_{X_0}(v_0)$ corresponding to quasi-singletons for ε. Let s_1 be the number of vertices removed. Extract corresponding reduced dataset $X_1 \subseteq X_0$, as well as $\mathrm{ch}_{X_1}(v_0)$, $E_{X_1}(v_0) = \mu_{X_1}(v_0)$, and $k_1 := k - s_1$.

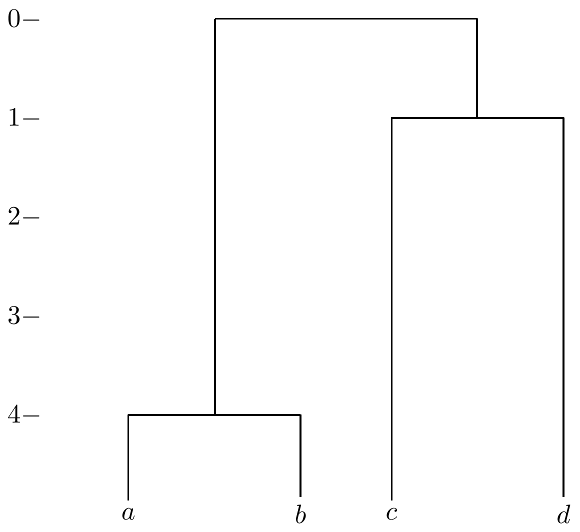

Figure 3. Dendrogram with quasi-singleton $\{a,b\}$ for $p^{-4}<\varepsilon\le p^{-1}$.

Step N. Assume that in the previous step, we are given a quadruple of families

$$(\mathscr{V}_{N-1},\mathscr{X}_{N-1},E_{N-1},\mathscr{K}_{N-1})$$

of sets $V\in\mathscr{V}_{N-1}$ of vertices in $D(X)$, datasets $X(V)\in\mathscr{X}_{N-1}$, an energy value $E_{N-1}=E_{X(V)}(V)$, and numbers $k_{N-1}(V)\le k$ (where $V\in\mathscr{V}_{N-1}$). Remove for all $V\in\mathscr{V}_{N-1}$ from $D(X(V))$ all vertices in $\mathrm{ch}_{X(V)}(v)$ corresponding to $s_N(v)$ quasi-singletons, where $v\in V$. Find all $V\in\mathscr{V}_{N-1}$ and $v\in V$ such that

(1) $k_{N-1}(V)-s_N(v)\ge 0$, and
(2) $E_{X(V)}(W_v)<E_{N-1}$ is smallest possible,

where $W_v:=\mathrm{ch}(v)\cup V\setminus\{v\}$. Extract corresponding quadruple of families

$$(\mathscr{V}_N,\mathscr{X}_N,E_N,\mathscr{K}_N)$$

of new vertex sets W_v, reduced datasets $X(W_v)\subseteq X(V)$, energy value $E_N=E(W_v)$, and $k_N(W_v):=k_{N-1}(V)-s_N(v)$.

Output. A list of clusterings consisting of quasi-singletons for ε and clusters produced above by collecting the remnants in each step.

Remark 4.14. *The output clusterings of Algorithm 4.13 all have energy of the form*

$$E+O(p^\alpha),$$

where E is independent of the clustering, and $\alpha<0$ is small.

We can now put things together in order to find clusterings in different ways:

Algorithm 4.15 ((Quasi-)Verticial split-LBG$_p$)**.** *Input.* As in Algorithm 4.6 (resp. Algorithm 4.13).

Step 1. Perform Algorithm 4.6 (resp. Algorithm 4.13).

Step 2. Perform Algorithm 4.9 for each cluster occurring in each clustering given out in the previous step.

Output. A list $(\mathscr{C}_i, (\mathbf{a}_j^{(i)})_{j\in J})_{i\in I}$ of E-suboptimal clusterings with corresponding list of E-center vectors $(\mathbf{a}_j^{(i)})_{j\in J}$ for clustering $\mathscr{C}_i$.

Both subroutines, Algorithms 4.6 and 4.9, boil down to counts and evaluations of $\mu(v)$ for vertices v. Therefore, we remark:

5. Dependence on the choice of the prime p

A natural issue is, how the outputs of the algorithms introduced in the previous sections depend on the choice of the prime number p. We will prove a finiteness result.

Recall that the energy of a verticial cluster C_V is of the form

$$E(C_V) = A \cdot p^{-\nu} \tag{10}$$

with natural numbers A and ν, and is additive on disjoint unions of clusters. Splitting a cluster is performed by replacing vertex v by the vertex set $\mathrm{ch}(v)$, and the change in energy is given by

$$E_{\text{new}} = E_{\text{old}} - E(C_v) + E(\mathrm{ch}(v)),$$

i.e. the difference is

$$\delta_v E_p := E(C_v) - E(\mathrm{ch}(v)).$$

Our approch towards minimising E_p is to refine the given clustering in the direction of largest $\delta_v E_p$. Now, the quantity $\delta_v E_p$ depends on the prime number p as shown by (10). This means that different p can result in different rankings of the vertices by the order in which they are split. We call this the *p-ranking* of the vertices of $D(X)$.

Example 5.1. *Assume we want to find verticial clusterings of data*

$$X = \{x_1, \dots, x_{13}\}$$

having underlying dendrogram as in Figure 4. Consider the vertices a, b, c, d in the underlying rooted vertex tree as depicted in Figure 5. Then Table 1 shows the different p-rankings of these vertices for $p = 2$, 3 and 5.

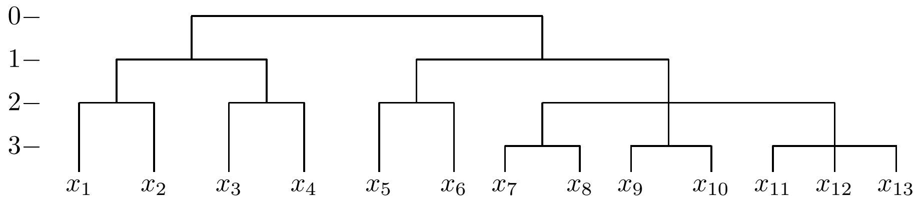

FIGURE 4. A dendrogram.

Theorem 5.2. *For all but finitely many primes, the p-rankings of the vertices of a given dendrogram $D(X)$ belonging to data X taken from a fixed p-adic field are the same.*

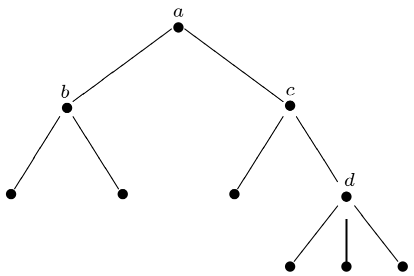

FIGURE 5. Vertex tree underlying Figure 4.

Rank	Vertex	$\delta_v E_2$
1.	a	$\frac{11}{2}$
2.	c	$\frac{9}{4}$
3.	b	2
	d	2

$p = 2$

Rank	Vertex	$\delta_v E_3$
1.	a	$\frac{22}{3}$
2.	c	$\frac{17}{9}$
3.	b	$\frac{7}{9}$
4.	d	$\frac{16}{27}$

$p = 3$

Rank	Vertex	$\delta_v E_5$
1.	a	$\frac{44}{5}$
2.	c	$\frac{44}{25}$
3.	b	$\frac{13}{25}$
4.	d	$\frac{26}{225}$

$p = 5$

TABLE 1. Vertex rankings for Figure 4.

Proof. The energy gradient for a vertex v can be written as

$$\delta_v E_p = P_v(t)|_{t=\frac{1}{p}}$$

for some polynomial $P_v(t)$ whose coefficients are natural numbers. By dividing off powers of t, we may assume that $P_v(t)$ has a non-zero constant term, hence that $P_v(0) > 0$. By the considerations from the previous sections, we know that

$$0 < P_v\left(\frac{1}{p}\right) < P_v(0) \tag{11}$$

for all primes p. By viewing $P_v(t)$ as a continuous function on the intervall $[0, 1/2]$, we see from the right inequality in (11) that $P_v(t)$ must be decreasing on some interval $[0, x]$ with positive $x \le \frac{1}{2}$ sufficiently small. It follows that the sequence of values $P_v\left(\frac{1}{p}\right)$ for prime $p \to \infty$ converges to $P_v(0)$. Since that limit equals $E(v)$ on the maximal subtree of $D(X)$ having v as its root, we have proven

$$\lim_{p\to\infty} P_v\left(\frac{1}{p}\right) = E(v).$$

In other words, for sufficiently large prime p, the vertex gradient can be approximated by the vertex energy. Hence the ranking of the vertices is approximatively the ranking of the numbers

$$\frac{E(v)}{p^{\ell(v)}}, \tag{12}$$

where $\ell(v)$ depends on the level of v in the dendrogram. The latter ranking does not change once p is sufficently large. Hence, for large p the vertex ranking does not change. □

Remark 5.3. *Notice from (12) that using a large prime number tends to force splitting vertices higher up in the hierarchy underlying the dendrogram. On the other hand, taking a small prime number allows to split also clusters containg lots of data at low levels in the hierarchy.*

Theorem 5.4. *Let $C \subseteq K$ be a cluster. If a is a center of C with respect to E_p for some prime p, then it is a center for all primes.*

Proof. From Lemma (4.7) it follows that

$$\epsilon_p(a) = E_p(C, \mathscr{C}, a) = \sum_{v\in V} \alpha_v \mu(v),$$

where V is the set of vertices on the path γ from the top v_0 down to a. As $\mu(v) = p^{-\ell(v)}$, and the $\ell(v)$ form a strictly increasing sequence $\ell_0, \dots, \ell_M$ of natural numbers as v proceeds along γ, it follows that $\epsilon_p(a)$ is given by evaluating the polynomial

$$F_\gamma(t) = \sum_{i=0}^{M} \alpha_i^\gamma t^{\ell_i}$$

in $t = \frac{1}{p}$, where $\alpha_i^\gamma > 0$ equals that number α_v with v such that $\ell(v) = \ell_i$. Now, $\epsilon_p(a)$ being a minimum means that in the collection

$$\{F_\gamma(t) \mid \gamma \text{ path } v_0 \rightsquigarrow X\}$$

the term $a_0^\gamma t^{\ell_0}$ is of lowest degree and that coefficient α_0^γ is smallest among those terms of lowest degree. And this does not depend on the choice of prime p. □

6. p-adic learning

In this section we discuss a learning situation in which some p-adic data $X \subseteq K$ together with a clustering $\mathscr{C}_X$ is used as a "training set". The idea is to classify new data Y taken from some p-adic field $L \supseteq K$ on the basis of X and $\mathscr{C}$. Without loss of generality we assume that the two p-adic fields L and K coincide.

6.1. p-adic classifiers.

Learning can be performed by using a classifier which integrates new data $y \in Y$ into an existing dendrogram $D(X)$ in order to find a suitable cluster for y. We will define such in the p-adic situation.

As it may happen that adjoining a point $y \in Y$ to X increases the size of the smallest p-adic disk containing the training data X, we use the point at infinity already introduced in [4]. This allows to classify those data in Y which cannot be classified on the basis of $(X, \mathscr{C}_X)$ as belonging to the "cluster at infinity". Our method will use the extended dendrogram

$$D_\infty(X) = D(\mathbb{P}(X)),$$

where $\mathbb{P}(X) = X \cup \{\infty\}$[2]. The datum ∞ will be depicted at the end of a path going upwards from v_0, whereas all other data will be at the end of paths leading downwards.

Example 6.1. *In Figure 6, some datum y is adjoined to a training dataset $X = \{a, b, c\}$. As it happens that the distance of y to X is larger than the diameter of X, the path $v_0 \rightsquigarrow y$ in the dendrogram $D_\infty(X \cup \{y\})$ has a portion going upwards in direction ∞.*

[2] Note that $D_\infty(X)$ is what is denoted by $D(X)$ in [4, 5].

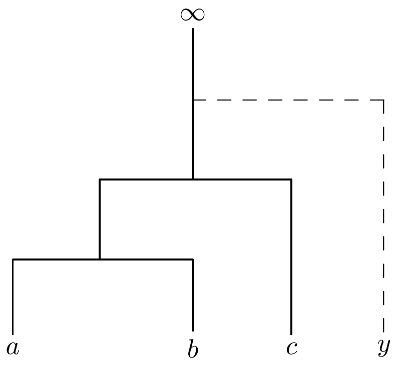

FIGURE 6. Dendrogram with cluster at infinity.

We call the pair $\mathcal{X} := (X, \mathscr{C}_X)$ a *classification* and have a *classification map*

$$\kappa_{\mathcal{X}} \colon \mathbb{P}(X) \to \mathscr{C}_X^\infty,\ x \mapsto C_x,$$

which assigns to each $x \in \mathbb{P}(X)$ the cluster C_x containing x, with

$$\mathscr{C}_X^\infty = \mathscr{C}_X \cup \{\{\infty\}\}.$$

Now, let $Z = X \cup Y$. We have the inclusion map $\iota \colon \mathbb{P}(X) \to \mathbb{P}(Z)$ which takes $x \in X$ to itself and ∞ to ∞.

Definition 6.2. *A p-*adic classifier for *Y* modeled on *$(X, \mathscr{C}_X)$ is a map*

$$\lambda \colon \mathbb{P}(Z) \to \mathscr{C}',$$

where $\mathscr{C}'$ is a clustering of $\mathbb{P}(Z)$, such that there exists an injective map $\phi \colon \mathscr{C}_X^\infty \to \mathscr{C}'$ making the diagram

$$\begin{tikzcd}
\mathbb{P}(X) \arrow[r, "\iota"] \arrow[d, "\kappa_{\mathcal{X}}"'] & \mathbb{P}(Z) \arrow[d, "\lambda"] \\
\mathscr{C}_X^\infty \arrow[r, "\phi"'] & \mathscr{C}'
\end{tikzcd}$$

commutative. The cluster $C_\infty := \lambda^{-1}(\phi(\{\infty\}))$ is called the residue *of λ. A classifier is called saturated, if ϕ is bijective.*

Remark 6.3. *Notice that ϕ is unique if it exists.*

Our first learning algorithm constructs the classifier sequentially by computing the distance to cluster centers for $\mathscr{C}_X$. Let $A = \{a_C \mid C \in \mathscr{C}_X\}$ be the set of given cluster centers $a_C \in C$. Then we have for $y \in Y$ the map

$$d_y \colon \mathscr{C}_X \to \mathbb{R},\ C \mapsto |y - a_C|_K\,,$$

and let $m_y := \min d_y(\mathscr{C}_X)$.

The vertex $v_y \in D_\infty(A \cup \{y\})$ nearest to y can be found e.g. using the p-adic expansions as given by (1). Namely, a vertex corresponds to a disk containing two or more p-adic numbers in $A \cup \{y\}$ having common initial terms determined by the radius of the disk. In geometric terms, traversing along the geodesic path $\gamma_y \colon \infty \rightsquigarrow y$ until all $a \in A$ have branched off γ_y yields the vertex v_y, and $\mu(v_y)$ is determined by the subset $C_{v_y} \subset A$ of those elements branching off precisely in v_y. The length of the path $v_0 \rightsquigarrow v_y$ gives m_y. And the map d_y is computed:

Lemma 6.4. *It holds true that*

$$\mathscr{C}_y := d_y^{-1}(m_y) = \left\{C_a \in \mathscr{C}_X \mid a \in C_{v_y}\right\}.$$

Proof. By what has been said above, the minimum is attained precisely for those clusters $C \in \mathscr{C}_X$ contained in C_{v_y}. Hence $C = C_a$ for some $a \in C_{v_y}$. □

The task is now to decide into which cluster from $\mathscr{C}_y$ to put y.

Algorithm 6.5. *Input.* A classification $\mathfrak{X}_0 := \mathfrak{X} = (X, \mathscr{C}_X)$, a set $A = \{a_C \mid C \in \mathscr{C}_X\}$ of cluster elements $a_C \in C$, and a set $Y \subseteq K$ of cardinality N.

Step 0. Set $C_\infty := \{\infty\}$.

Step 1. Take $y := y_1 \in Y$, and compute v_y, m_y, C_{v_y}, $\mathscr{C}_y$, and μ_{v_y}.
Case 1. If $C_{v_y} = \{a\}$, then set $C_y := C_a \cup \{y\}$ and $A_1 := A$.
Case 2. If $\# C_{v_y} > 1$, then find the subset $C^y \subseteq C_{v_y}$ of all elements whose nearest vertex in $D_\infty(C_{v_y} \cup \{y\})$ equals v_y. If $C^y = \emptyset$, then set $C_y = \{y\}$ and $A_1 := A \cup \{y\}$. Otherwise, find all elements $a \in C^y$ with minimal energy $E(C_a \cup \{y\})$. If there is more than one such a, then $C_y := \{y\}$ and $A_1 := A \cup \{y\}$. Otherwise, $C_y := C_a \cup \{y\}$, and $A_1 := A$.

In any case, produce $Y_1 := Y \setminus \{y\}$, A_1 and classification $\mathfrak{X}_1 := (X_1, \mathscr{C}_{X_1})$, where $X_1 = X \cup \{y\}$ and $\mathscr{C}_{X_1} := \{C_y\} \cup \mathscr{C}_X \setminus \{C_a\}$. Terminate, if $Y_1 = \emptyset$.

Step N. Assume that in the previous step, sets Y_{N-1}, A_{N-1} and a classification $\mathfrak{X}_{N-1}$ have been given out. Then perform Step 1 with $\mathfrak{X} := \mathfrak{X}_{N-1}$, $A := A_{N-1}$, and $Y := Y_{N-1}$.

Output. On termination in *Step M*, an optimal classifier

$$\lambda \colon \mathbb{P}(X_M) \to \mathscr{C}_{X_M},\ x \mapsto C_x,$$

modeled on $\mathfrak{X}_0$.

Proof of optimality. In each step N, $y_N \in Y_N$ is assigned to the cluster $C \in \mathscr{C}_{X_N}$ with minimal energy $E(C \cup \{y_N\})$. □

Theorem 6.6. *The outcome of Algorithm 6.5 does not depend on the choice of the set A of cluster representatives.*

Proof. The outcome of *Step* 1 does not depend on A. □

Remark 6.7. *A consequence of Theorem 6.6 is that Algorithm 6.5 does indeed effect learning in the sense, that to any $y \in Y$ is assigned a cluster depending on the already existing clusters. Representing a cluster by a single element makes learning efficient.*

6.2. **Adaptive learning.** During the learning process[3], it can become useful to subdivide big clusters of the extended dataset $X \cup Y$. This is not a problem, as the old cluster centers can be reused in the new clustering.

Lemma 6.8. *Let C be a cluster, and $a \in C$ a center of C. Assume that C' is a subcluster of C containing a, then a is a center of C'.*

[3] Or if for some reason one wants to perform a variation of split-LBG$_p$ in which centers are computed after each clustering step, instead of after termination of clustering.

Proof. Clearly, it holds true that

$$E_p(C,\mathscr{C},a) \le E_p(C,\mathscr{C},a'), \tag{13}$$

where $\mathscr{C} = \{C\}$ and $\mathscr{C}' = \{C\}$. Assume that $a' \in C'$ is a center of C'. Now, inequality (13) implies that

$$\begin{aligned} E_p(C',\mathscr{C}',a) + \sum_{x\in C\setminus C'} |x-a|_K &= E_p(C,\mathscr{C},a) \\ &\le E_p(C,\mathscr{C},a') \\ &= E_p(C',\mathscr{C}',a') + \sum_{x\in C\setminus C'} |x-a'|_K \end{aligned}$$

Since, by the cluster property of C', it holds true that

$$|x-a|_K = |x-a'|_K$$

for all $x \in C \setminus C'$, it follows that

$$E_p(C',\mathscr{C}',a) \le E_p(C',\mathscr{C}',a'), \tag{14}$$

and, because a' is a center of C', this yields an equality in (14) i.e. a is a center of C'. □

Remark 6.9. *Notice that Lemma 6.8 does not hold true, if we allow C' to be an arbitrary subset of C. E.g. assume in Figure 7 that $C = \{a,b,c,d\}$. Then a is a center of C, as can be verified from the left dendrogram. However, a is not a center of $C' = \{a,c,d\}$, as the right dendrogram reveals. Namely, in the first case, we compute with $\mathscr{C} = \{C\}$ and $\mathscr{C}' = \{C'\}$:*

$$\begin{aligned} E(C,\mathscr{C},a) = E(C,\mathscr{C},b) &= |a-b|_K + 2\cdot|a-c|_K \\ &< |c-d|_K + 2\cdot|a-c|_K = E(C,\mathscr{C},c) = E(C,\mathscr{C},d), \end{aligned}$$

and in the second case:

$$\begin{aligned} E(C',\mathscr{C}',c) &= |a-c|_K + |d-c|_K \\ &< 2\cdot|a-c|_K = E(C',\mathscr{C}',a). \end{aligned}$$

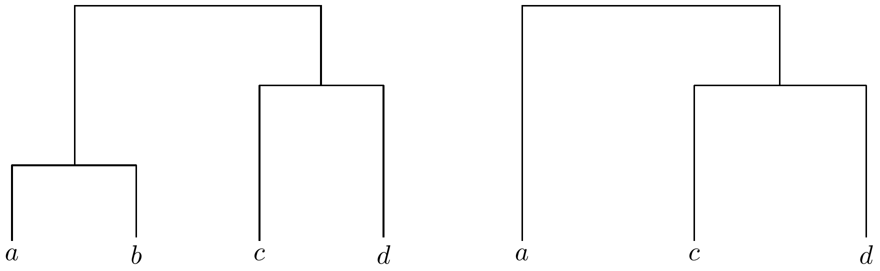

FIGURE 7. Dendrogram and subdendrogram.

At last, we propose the splitting of high-energy clusters accumulated during the learning process:

Algorithm 6.10. *Input.* $r \geq 0$. Otherwise, as in Algorithm 6.5.

Perform Algorithm 6.5 with modification:

Step N'. Perform *Step* N. If for $y := y_N$ it holds true that $E(C_y) > r$, then split cluster C_y into its maximal proper subclusters, and adjoin to A_N new cluster centers using Algorithm 4.9.

7. Conclusion

A straightforward translation of the split-LBG algorithm to the situation of classifying p-adic data does not exist. However, if clusterings, cluster centers and their numbers are allowed to vary, then the minimisation problem for the p-adic energy functional defined by distances to centers does make sense. Sub-optimal algorithmic solutions to the minimisation problem are presented, in which the choice lies in whether or not to remove in each step quasi-singletons, i.e. clusters which are almost singletons because of their energy values being lower than a given threshold. The method is to find rankings of vertices in the dendrogram associated to the p-adic data. The outcome depends on the prime number p, but it is shown that for all but finitely many primes the rankings are identical. The consequence for applications to data anlaysis is that for fixed prime p, the classification results do not depend on the p-adic representation of the data, as long as the dendrograms are isomorphic. Furthermore, the minimising property for given cluster centers holds true independently of the prime. This means that if some datum is a cluster center for one prime, it is a cluster center for all primes (for which the corresponding cluster is not larger). Using p-adic cluster centers, one can construct classifiers from given clusterings. This can be applied to learning situations.

Acknowledgements

The author acknowledges support from the DFG-project BR 3513/1-1. Andrei Khrennikov is warmly thanked for directing the author's attention to [1] where the issue of a p-adic form of split-LBG is raised. Thanks to Fionn Murtagh for remarks leading to improving the exposition of the article. The anonymous referee is thanked for pointing out various inaccuracies and errors. The Institut für Photogrammetrie und Fernerkundung at University Karlsruhe is thanked for the opportunity to write down this article.

References

[1] J. Benois-Pineau, A.Yu. Khrennikov, N.V. Kotovich. *Segmentation of Images in p-Adic and Euclidean Metrics.* Dokl. Math., 64, 450–455 (2001)

[2] J. Benois-Pineau and A.Yu. Khrennikov. *Significance Delta Reasoning with p-Adic Neural Networks: Application to Shot Change Detection in Video.* The Computer Journal. In Press. DOI: 10.1093/comjnl/bxm087.

[3] Patrick Erik Bradley. *An ultrametric interpretation of building related event data.* Preprint.

[4] Patrick Erik Bradley. *Degenerating Families of Dendrograms.* J. Classif., 25, 27–42 (2008)

[5] Patrick Erik Bradley. *Mumford dendrograms.* The Computer Journal. In Press. DOI: 10.1093/comjnl/bxm088.

[6] Andrei Khrennikov and Brunello Tirozzi. *Algorithm of Learning of p-adic Neural Networks.* Preprint.

[7] Yoseph Linde, Andrès Buzo, Robert M. Gray. *An Algorithm for Vector Quantizer Design.* IEEE Trans. Commun., 28, 84–95 (1980)

[8] Fionn Murtagh. *On ultrametricity, data coding, and computation.* Journal of Classification, 21, 167–184 (2004)

[9] Fionn Murtagh. *From Data to the p-Adic or Ultrametric Model*. p-Adic Numbers, Ultrametric Analysis and Applications, 1, 53–63 (2009)

[10] Christophe Perruchet. *Hierarchical Classification of Mathematical Structures.* Stat. Prob. Lett., 1, 61–67 (1982)